John von Neumann Institute for Computing

Parallel Global Optimisation for Oil Reservoir Modelling

S. Gómez, N. del Castillo

published in

Parallel Computing:
Current & Future Issues of High-End Computing,
Proceedings of the International Conference ParCo 2005,
G.R. Joubert, W.E. Nagel, F.J. Peters, O. Plata, P. Tirado, E. Zapata (Editors),
John von Neumann Institute for Computing, Jülich,
NIC Series, Vol. **33**, ISBN 3-00-017352-8, pp. 261-268, 2006.

http://www.fz-juelich.de/nic-series/volume33

Parallel Global Optimisation for Oil Reservoir Modelling *

Susana Gomez[a], Nelson del Castillo[a]

[a] IIMAS, Univ. Nacional A. de Mexico, Mexico, susanag@servidor.unam.mx

1. Introduction

The objective of this work is to develop a robust mathematical and computational tool to forecast the production of an oil reservoir at a reasonable computational cost. To forecast production it is neccessary to simulate the flux in a porous media, to obtain the pressure and saturation (state variables), time and space dependent. This implies the solution of a highly complex system of PDEs, which has been thoroughly studied. There are today, a set of commercial simulators that incorporate the latest numerical efficient and robust methods, that guarantee the existence and uniqueness of the solution and the desired precision. Here we will use one of this simulators, called ECLIPSE, able to handle a wide espectrum of reservoir conditions (3-D, multiphase flow, compositional, etc).

This system depends linearly or non-linearly, on a set of coefficients or parameters p, that are space dependent, related to the porous media characteristics (transmissibility and porosity) of the site under study. In practice, these parameters are not known, although some geological and geophysical information might be avalilable. However, the high sensitivity of the simulation and forecast (simulation for future periods of time) to the knowledge of these characteristics p, makes their determination the main and most difficult problem. To be able to find these coefficients, using limited historical data on the state variables measured at the wells, a data-fitting least-squares optimisation problem can be solved [9]. This inverse problem is generally known as History Matching characterisation.

Although the solution of the flux equations is well-posed in the Hadamard sense, the inverse parameter identification problem is not. This implies that uniqueness of the optimal solution is not guaranteed unless more a-priori information on the solution is known and incorporated in the problem definition. The continuity of the solution to the data is neither satisfied here (the inverse operator may not be continuous), as a small error in the measured data, and/or in the numerical solution of the direct problem, can produce optimal solutions completely misleading (see for instance [12,10]). For this continuity problem, some regularisation method has to be used to be able to get the best possible stable approximation to the solution. Also, the evaluation of the objective function (at least once at each iteration of the optimization process), implies the solution of the PDEs system, making the History Matching a highly computationally intensive problem.

Then, considering the non-convexity of the objective function, and taking into account the non-continuity problem, the goal here is to get a set of stable optimal solutions with good match to the data, in reasonable computer time. Thus, it is neccessary to develop a robust, reliable and efficient global optimisation method. We then solve the flux equations for each optimal solution (a set of coefficients) at future time periods, to produce a set of scenarios of production forecast.

Here, we work with the Parallel Tunneling Global Optimisation Method (PTGOM) [8], that finds a set of local optima with good match to the data. This method requires the use of a gradient based local optimisation method, and thus its speed and robustness depend partly on the effectiveness of the local method. In this work, we also test the ability of the local method used, the Truncated Gauss Newton with Conjugate Gradient as the linear solver, to regularise the optimal local search. In a

*This work has been partially supported by UNAM-PAPIIT, GRANT No. IN-112999, CONACyT GRANT No.35458-A and by Consortium OPTIMA (Total and Agip).

future publication we give comparison results when a Limited Memory Quasi-Newton method is also used.

Although PTGOM is deterministic in its general design, in the present version it makes a space search starting from random directions, to locate points in other valleys and this gives the possibility to parallelise them. Other stochastic methods have been used to solve the history matching problem, but their computational cost is still an open issue. Numerical results will be given here for a benchmark problem called PUNQ-S3.

2. Global Optimisation

In order to introduce notation, we will call $Mod(p)$ the solution of the flux system of PDEs, for a given value of the porous media characteristics or parameters $p = g(x, y, z)$. The state variables, which are space and time dependent $sv(x, y, z, t)$, are the bottom-hole presure BHP, the water-cut WCT and the gas-oil ratio GOR.

To find p, we have to solve an inverse problem: Given a limited set of measurements $data_{i,j,k}$ of the state variables sv mentioned above, measured at the observation wells at several time intervals (where i is the well number, j is the measurement number at each well and k is the time interval), find the parameters p that produce $Mod(p) \approx data_{i,j,k}$ as accurately as possible.

This inverse problem (given the effect sv find the cause p), can be posed as a least-squares data fitting optimisation problem,

$$\min \qquad F(p) = \frac{1}{2} \parallel f(p) \parallel_2^2, \quad f(p) = Mod(p,t) - data_{i,j,k}, \tag{1}$$

where $f : \Re^n \rightarrow \Re^m$, and $f(p)$ is the residual vector. Some a-priori information on the solution, based on geological knowledge of the site under study, allow us to impose some bound constraints on the parameter values

$$p_{min} \leq p \leq p_{max}. \tag{2}$$

In order to get a set of stable local optimal solutions with good match to the data, a global optimisation method has to be used, with certain characteristics that most well known methods fail to have. A more detailed description of these characteristics will be given in a future publication.

In this work, we will use the Tunneling Methods and exploit their stochastic element to design a parallel version. A brief description of these methods is given now, and the reader is referred to [15,14,1].

2.1. The Sequential Tunneling Method

The method was designed to find the global minimum x_G^* of a general function $F \in C^2$, with bound constraints and/or general non-linear constraints [14].

The basic deterministic idea of the Method is to find a local minimum x^* with $F(x^*) = F^*$ during a minimisation phase, using a local bounded optimisation method (gradient based), and then during a tunneling phase, find a point x_{tu} in another valley with $F(x_{tu}) \leq F^*$. This inequality implies that valleys whose minimum is above the level F^* will not be found. Once in another valley, starting from x_{tu}^*, it finds again another local minimum, ending with a sequence of local optimal solutions with decreasing function values $F(x_1^*) \geq F(x_2^*) \geq ... \geq F(x_G^*)$, and ignoring all the local minima with larger values than the best already found. These two phases are repeated alternatively until convergence to the global solution or until a prefixed number of function evaluations or CPU time, is achieved.

2.1.1. Tunneling Phase

Once a local minimum has been obtained, to be able to tunnel from one valley to another using gradient-type methods, it is necessary first to destroy the minimum, placing a pole at the last minimum point x^* in such a way that at $x = x^*$ the transformed function $T(x)$ is infinite and that $T(x) \leq 0$ at points $x \neq x^*$ whenever $F(x) \leq F^*$. To place a pole that satisifes these conditions we use here the Exponential Tunneling Function [1].

$$T(x) = (F(x) - F^*) \exp\left(\frac{\lambda^*}{\parallel x - x^* \parallel}\right) \leq 0. \tag{3}$$

An alternative Tunneling function (that we call classical) can be found in [15,14]. Parameter λ^* is the strength of the pole and plays an important role. The tunneling function $T(x)$ inherits the multimodality of the original function $F(x)$, and the local method used to solve problem 3, may find a critical point of $T(x)$. If λ is taken large enough, a smoothing effect takes place (removing critical points of $T(x)$) although in practice this large value of λ, that depends on the first and second derivatives of F, cannot be computed explicitly. Also, the larger λ is, the wider the pole becomes, and a local valley within that neiborhood of x^* might not be found. The method then, would take the smallest λ that gives a descent direction for $T(x)$ and the use of mobile poles to deal with the critical points of $T(x)$, allows the local method to find the solution of problem 3. A detailed description on the implementation of the method and the way to handle mobile poles and relative tolerances, can be found in [4].

To solve inequality problem 3, descent directions are generated with the same local algorithm used in the minimisation phase, with appropriate stopping conditions to check convergence for problem 3.

This method has also some convenient properties not shared by most other global methods: it is able to find minima at the same level, with equality satisfied in 3. But in order to avoid finding again minima already found at the same level, the poles at the minima already found have to be active, and the Tunneling function has to be modified in the following way

$$T(x) = (F(x) - F^*) \prod_{i=1}^{l} \exp\left(\frac{\lambda_i^*}{\parallel x - x_i^* \parallel}\right) \leq 0. \tag{4}$$

In applications like the one we solve in this work, where a set of minima with small residuals is sought, the so called minima at the same level can be defined to a certain tolerance and thus the searched points in other valleys would satisify the modified inequality $F(x_{tu}) \leq F^* + TOL$. This tolerance may be given by the expected value of the residual $f(x)$.

2.2. Parallel Design

To start the seach for x_{tu} that satisfies $F(x_{tu}) \leq F^*$ through 3, we need to take an initial point x^0 in a neighborhood of the last local minimum found x_i^*. In this work we have taken this initial point along random directions

$$x^0 = x_i^* + \epsilon r$$

where ϵ depends on the scale and the needed precision of the problem (see [4]), and r is a random direction within $r \in [-1, 1]$. From this initial point the search for the solution of inequality problem 3 starts, using the same method we used to solve the minimisation problem to generate descent directions and step lengths. If the search from a prefixed number of initial points (in different directions)

in the neighborhood of x_i^* has been unsuccessful, the search continues starting from points taken in the whole feasible region until a maximum prefixed number is reached.

The efficiency of the search for points in other valleys after a new minimum has been found, depends strongly on which direction r the initial point $x \neq x^*$ for the tunneling phase is taken. Thus the idea that supports the parallelisation of the method, is to allow each processor to start the search in a different direction. This is specially important when the number of unknown parameters or when the number of local optima are large, as the search in the subspace defined by the level set $f(x) \leq f^* + TOL$, has to be exhaustive.

However, in order to make the parallel process efficient, avoiding too many message passing, each processor will carry out independently both phases: the tunneling phase to search for points in other valleys, and if no one else has found a better minimum, once the point x_{tu} is found, it continues the search for the local minimum of that valley, by performing the minimisation phase.

A master processor has to control the process, so it finds the first local minimum starting from a given initial set of parameters, obtained using geological information of the site and sends this first minimum to all other processors. Then, it receives the new minima found by any processor, checks that it is different that the ones already reported, in which case it sends it to all other processors. It also checks for the stopping conditions, that in this application means the CPU time or a fixed number of local minima to be found determined a-priori. It sends a message to all processors to finish the run, generates the output files and stops.

The slave processors, will check for new messages concerning new minima, only while performing a tunneling phase, that will continue now from the new minimum received, creating a pole to destroy it and updating the function value F^*. This phase proceeds now to find a point x_{tu} with $F(x_{tu}) \leq F^* + TOL$. While in the minimisation phase, the slave processor only checks the general stopping messages.

2.3. Local optimisation, Regulatisation and Scaling

In this work the local optimisation method we are using is a modified version of the code TRON (see [16]) based on a Truncated Newton Method with a trust region strategy for global convergence. Here we test the Gauss-Newton approximation $H = J^T J$ to the Hessian, where J the Jacobian matrix is given by the simulator. The Gauss-Newton system of linear equations, used to find a descent direction s_k at the non-linear iteration k, is solved using a Conjugate Gradient Method.

During the History-Matching, the Hessians may be ill-conditioned due to the ill-posednes of the parameter estimation inverse problem. One of the consequences is that the valleys of the non-convex objective function may become very flat, and convergence to the local optimal solutions may not be achieved with enough precision. Scaling by variable transformation converts the variables from units that typically reflect the physical nature of the problem to units that display certain desirable properties during the minimisation process. For global optimisation, the shape of the valleys of the non-convex objective function can be modified through scaling [11], as although the objective function values would not change, the gradient and Hessian of the problem will be transformed. If the minimisation problem has simple bounds on the variables, as is the case here, say $x_i \in [l_i, u_i]$ for $i = 1, ...n$, the new scaled variables $y_i \in [l_i^{new}, u_i^{new}]$ can be defined as

$$y_i = \left(\frac{u_i^{new} - l_i^{new}}{u_i - l_i} \right) x_i + \frac{u_i l_i^{new} - u_i^{new} l_i}{u_i - l_i} u_i - l_i \tag{5}$$

This transformation [7] can be written in matrix form as $x = Dy + c$ where D is a diagonal matrix and c is a vector. Although the function in the scaled variables $h(y)$ is not altered $h(y) = F(x)$, the

derivatives of the objective function are scaled. Let g_y and H_y denote the gradient vector and the Hessian matrix of the scaled variables, and g and H the original ones. These derivatives are then related by

$$g_y = Dg \qquad \text{and} \qquad G_y = DGD$$

Hence, even a "mild" scaling may have a substantial effect on the Hessian, and this in turn may significantly alter the convergence rate of the optimisation algorithm. The linear system becomes preconditioned by matrix D.

Due to the non-continuity of the solution to the data, a small perturbation of the data (measurements errors), can cause a large perturbation of the solution. To prevent this error propagation, the degree of approximation to the solution has to be limited and regularisation methods have to be used to balance the degree of approximation (the regularisation parameter) and the error propagation. Here, we study the regularisation properties of iterative methods [13], by stopping the Conjugate Gradient iteration used to solve the Gauss-Newton linear system that finds the descent direction. As the Hessian approximation $H = J^T J$ may be ill-conditioned, it is necessary to stop the iteration before the error of the data propagates. We use here the triangle method (see [5], where it was applied for linear systems with errors) to find the corner of the L-curve generated with the Conjugate Gradient iterations, to determine automatically the stopping iteration (which becomes the regularisation parameter). This parameter re-defines the size of the trust region for the next Gauss-Newton iteration. Also, as the scaling described before implies a diagonal preconditioner of the Gauss-Newton system, it may have a regularisation effect that will also be tested here.

3. Application to History Matching of Oil Reservoirs

The PUNQ-S3 problem [17] has recently become quite popular as a sort of benchmark for history matching and risk-analysis methodologies. It is a dynamical reservoir model based on a real west Africa field, which has been discretised using a 19x28x5 corner-point grid, with 1,721 active cells. Strong water supports come from north and west, while two faults close the reservoir at east and south, and a small gas cap is present at the top of the formation. A history period, simulating 8 years of production from six wells located close to the gas-oil contact (GOC), was generated by The Nederland Organisation for Scientific Research (TNO) using geostatistical distributions of porosities and permeabilities. Gaussian noises have been added to the collected well data to reproduce a real measurement process. Then, 8 years of forecast with five additional infilling wells have been simulated. The data set, consisting of noisy well-data, grid structure, transmissibility and porosity distributions, is available at the TNO web site.

The complexity of the parameter identification problem was avoided in this work by adopting a set of parameters, based on Gradzone Analysis [2], obtained by the TOTAL Geoscience Research Centre (GRC). In this work, we used the aforementioned gradzones in which the available a priori geological information was also included in the analysis [3]. For PUNQ-S3, this analysis leads to a vector of 10 history matching parameters p, 5 porosities and 5 transmissibilities , which contains a signature of the geological model. One multiplier for each property was assigned to every layer. To restrict the evolution of the system into a physically reasonable region, simple bounds, acting as perfectly absorbing surfaces, are imposed to the parameters as follows $0.1 \leq p_i \leq 3, \quad i = 1, ...10.$

4. Numerical Results

All the experiments were run on a Beowulf cluster with 8 nodes with two pentium III 1 Ghz at each node. Communication is performed with ethernet at 100Mbps, and Linux system. In this section we show a limited set of results and a complete discussion and graphs will be given in a future publication. In Figures 1 and 2, the different scenarios obtained with each local minima (min1, min2 etc.) with good match to the data are shown for two different wells, over 16 years. The first 8 years show the history matching and the next 8 are the forecast results. In the graphs, the real data is referred as *data*. In Figures 3, 4 and 5, the same set of scenarios is presented for the whole field, for water, gas and oil. In Figure 6, the important effect of scaling (sca) and regularizing (ilcu)is shown. In Figure 7 the graph shows the computer time behaviour when 1,3,5 and 7 processors were used. In Table 1 we show the results comparing the sequential versus the parallel versions. The latest is 17 times faster. Finally in table 2 the speedup and the efficiency of the parallel implementation is given.

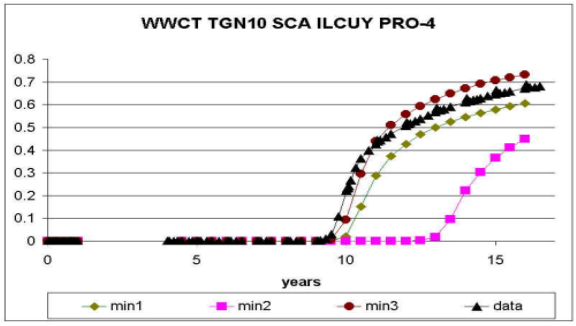

Figure 1. Scenarios produced by 3 different minima and the real data, at Well 4

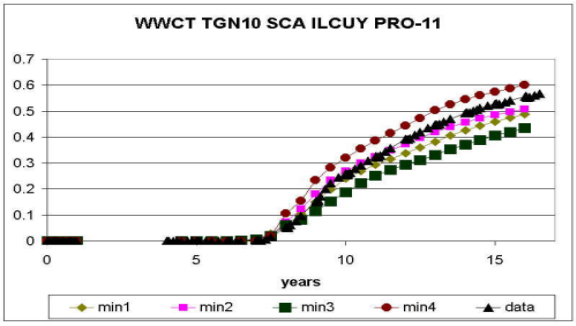

Figure 2. Scenarios produced by 4 different minima and the real data, at Well 11

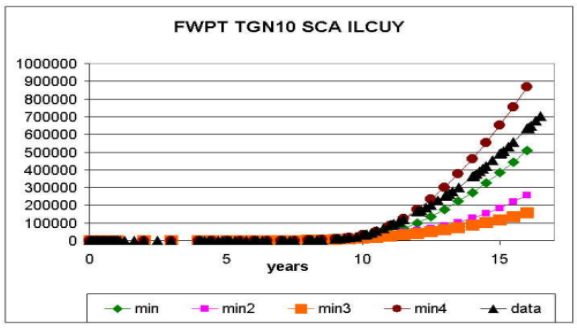

Figure 3. Field H-M and Forecast.Water

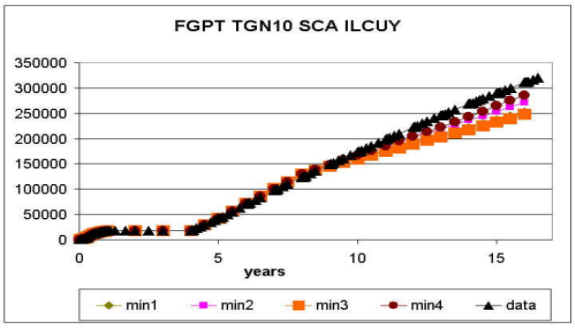

Figure 4. Field H-M and Forecast. Gas

	F^*	No. Local Min	No. Function Eval.	Time(hours)
Tun-TGN-Seq	0.2884	3	1026	17:35
Tun-TGN-Par	0.2734	4	67	1:10

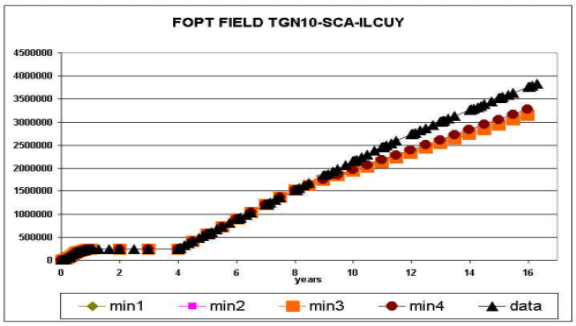

Figure 5. Field H-M and Forecast.Oil

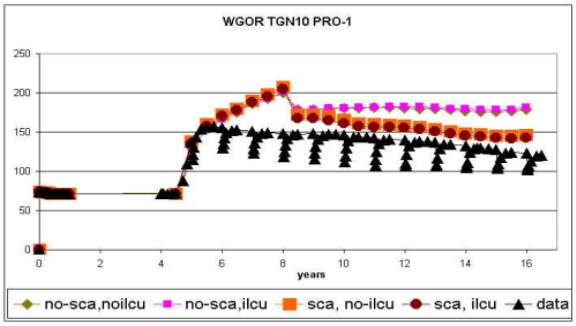

Figure 6. Good effect of Scaling (sca) and Regularising (ilcu)

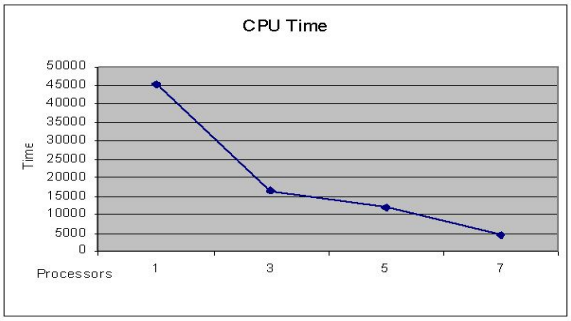

Figure 7. Parallel CPU Time

Table 1. Sequential versus Parallel Mehtods.

Processors	CPU Time (sec)	Speedup	Efficiency
1	45180		
3	16292.91	2.7730	0.9243
5	11743.47	3.8472	0.7694
7	4150.54	10.8853	1.5550

Table 2. Parallel TUNNEL-TGN behaviour.

5. Conclusions

The need to generate a set of optimal solutions has been shown, as the last minimum, with the lowest value of the objective function, may not give the best approximation. It is the set of solutions that gives the neccessary information for decision taking. The same is true for each individual well and for the field forecast.

The parallelisation of the method accelerated the procces 17 times, presenting the Parallel Tunneling Method as a promising toool for real History Matching applications.

Acknowledgements. We want to thank Schlumberger Mexico, for letting us use a multiple ECLIPSE license, on a cluster machine. Also, we thank the Supercomputo department at DGSCA UNAM for the use of their parallel machines to test the parallelisation scheme. We acknowledge the help we received from Francisco J. Cardenas to be able to run ECLIPSE on the cluster he supervises.

References

[1] Barron, C. and Gomez, S., *The Exponential Tunneling Method*, Reporte de Investigacin IIMAS, Vol. 1, No. 3, 1991.

[2] Bissel, R.C., *Calculating Optimal Parameters for History Matching*, Proc., 4th European Conference on Mathematics of Oil Recovery, Rros, Norway, 7-10 June 1994.

[3] Brun, B., Gosselin, O., and Barker, J.W., *Use of Prior Information in Gradient-based History Matching*, paper SPE 66353 presented at the 2001 SPE Reservoir Simulation Symposium, Houston, 11-14 February 2002.

[4] Castellanos, L. and Gomez, S., *A New Implementation of the Tunneling Methods for Bound Constrained Global Optimisation*, Reporte de Investigacin IIMAS, Vol. 10, No. 59. pp. 1-18, 2000.

[5] Castellanos, L., Gomez, S. and Guerra, V., *The Triangle Method to Locate the Corner of the L-Curve*, Journal of Applied Numerical Mathematics, Vol. 43, No. 4. pp. 359-373, 2002.

[6] Giliazov S.F. and Goldman N.I., *Regularisation of ill-posed problems by iteration methods*, Kluwer Academic Publishers, 2000.

[7] Gill, P., Murray,W. and Wright, M.H., *Practical Optimisation*, Academic Press, 1981.

[8] Gomez, S., Del Castillo, N., Castellanos, L. and Solano, J., *The Parallel Tunneling Method*, Parallel Computing Vol. 29, pp. 523-533, 2003.

[9] Gomez, S., Gosselin, O. and Barker, J., *Gradient-Based History-Matching with a Global Optimisation Method*, SPE Journal, pp. 200-208, June 2001.

[10] Gomez, S., Ono, M., Gamio, C. and Fraguela, A., Reconstruction of Capacitance Tomography Images of Simulated Two Phase Flow Regimes, J. of Applied Numerical Mathematics, Elsevier, Vol. 46(6), pp. 511-518, 2003.

[11] Gomez, S., Solorzano, J., Castellanos, L. and Quintana, M.I., *Tunnelling and Genetic Algorithms for Global Optimisation*, in . Hadjisavvas and Pardalos (eds.), Advances in Convex Analysis and Global Optimisation. Non Convex Optimisation and its Applications, Kluwer Academic Pub. pp. 553-567, 2001.

[12] Gomez, S., Perez, A., Dilla, F and Alvarez, R.M., *On the Automatic Calibration of a Confined Aquifer*, Computational Methods in Water Resources, XIV, Kluwer Academic Pub., 2002.

[13] Kelley C.T., *Iterative methods for linear and nonlinear equations*, Frontiers in Applied Mathematics, Vol. 16, SIAM, 1995.

[14] Levy, A. and Gomez, S., *The Tunneling Method Applied to Global Optimisation*, in P.T. Boggs, R.H. Byrd and R.B. Schnabel (eds.), Numerical Optimisation, SIAM, pp. 213-244, 1985.

[15] Levy, A. and Montalvo, A., *The Tunneling Algorithm for the Global Minimisation of Functions*, SIAM J. Sci. Stat. Comput., Vol. 6, No. 1, pp. 15-29, Jan. 1985.

[16] More, J.J. and Chih-Jen, L., *Newton's Method for Large-Scale Bound Constrained Optimisation Problems*, SIAM Journal on Opt. Vol. 9, No. 4, pp. 1100-1127, 1999.

[17] *The PUNQ Project*, URL: www.nitg.tno.nl/punq/